

Corrigé partiel Calculabilité et Complexité

22 novembre 2020

Dans ce corrigé, on note par $N[u]$ l'ensemble contenant u ainsi que tout ses voisins.

Question 1. Soit $G = (V, E)$ un graphe. On remarque que pour qu'un ensemble dominant D soit non-ambigu, il faut que $N[u] \cap N[v] = \emptyset$ pour toute paire de sommets $u, v \in D$. En particulier, $\sum_{u \in D} |N[u]| = |V|$. On montre que le cycle de taille n possède un ensemble dominant non-ambigu si et seulement si $n \equiv 0 \pmod{3}$, i.e. si n est multiple de 3.

Si n est multiple de 3, on peut construire un ensemble dominant non-ambigu D en prenant les sommets $1, 4, 7, 10, \dots$, i.e. les sommets d'indice $1 \pmod{3}$. Si n n'est pas multiple de 3, il faut montrer que G n'a pas de dominant non-ambigu. Comme n n'est pas multiple de 3, on a $n = 3k + 1$ ou $n = 3k + 2$ avec $k \geq 1$. Comme un sommet ne domine que lui et ses deux voisins (soit trois sommets au total), k sommets ne suffisent pas à dominer G . Ainsi tout ensemble dominant (qu'il soit non-ambigu ou pas) a taille au moins $k + 1$. Mais on a alors que $\sum_{u \in D} |N[u]| \geq 3(k + 1)$, ce qui est strictement plus grand que n , car $n \in \{3k + 1, 3k + 2\}$. Ainsi, au moins un sommet de G est dominé plus d'une fois, et donc D n'est pas un dominant non-ambigu.

Question 2. Soit $G = (V, E)$ un graphe. Un certificat d'une instance positive est un sous-ensemble de sommets $D \subseteq V$. Sa taille est donc $O(|V|)$. L'entrée du vérificateur est le graphe $G = (V, E)$, un entier k , et un sous-ensemble de sommets $D \subseteq V$. Le vérificateur doit répondre **Vrai** si et seulement si D est un ensemble non-ambigu de G de taille au plus k . Voici l'algorithme :

Algorithme 1 Vérificateur

Entrée Un graphe $G = (V, E)$, un entier k , un sous-ensemble de sommets $D \subseteq V$

Sortie **Vrai** si et seulement si D est un ensemble dominant non-ambigu de G

```
1: Si  $|D| > k$  alors
2:   Retourner Faux
3: Fin Si
4: Pour tout sommet  $u \in V$  faire
5:    $\text{cpt} \leftarrow 0$ 
6:   Si  $u \in D$  alors
7:      $\text{cpt} \leftarrow \text{cpt} + 1$ 
8:   Fin Si
9:   Pour tout  $v \in \text{Adj}[u]$  faire
10:    Si  $v \in D$  alors
11:       $\text{cpt} \leftarrow \text{cpt} + 1$ 
12:    Fin Si
13:   Fin Pour
14:   Si  $\text{cpt} \neq 1$  alors
15:     Retourner Faux
16:   Fin Si
17: Fin Pour
18: Retourner Vrai
```

On peut représenter l'ensemble D par un tableau de booléens tab avec $\text{tab}[u] = \text{Vrai}$. Ainsi, vérifier que $|D| \leq k$ se fait en temps $O(|V|)$ et vérifier si un sommet $u \in V$ appartient à D se fait en temps $O(1)$.

L'algorithme parcourt au plus une fois chaque sommet, et deux fois chaque arête. La complexité est donc $O(|V|) + O(|V| + |E|) = O(|V| + |E|)$, ce qui est linéaire en la taille du graphe.

Question 3. Le problème EDNA possède un vérificateur polynomial d'après la question 2. On en déduit que EDNA appartient à NP.

Question 4. Voici un algorithme qui résout le problème EDNA. Cet algorithme teste tous les sous-ensembles de sommets de taille au plus k et vérifie s'il s'agit d'un ensemble dominant non-ambigu ou non en appelant le vérificateur polynomial de la question 2.

Algorithme 2 EDNA

Entrée Un graphe $G = (V, E)$, un entier k

Sortie Un ensemble dominant non-ambigu de taille au plus k s'il existe, \emptyset sinon.

```

1: Pour  $i$  allant de 1 à  $k$  avec un pas de 1 faire
2:    $\mathcal{S} = \{\text{sous-ensembles de } V \text{ de taille exactement } i\}$ 
3:   Pour tout  $D \in \mathcal{S}$  faire
4:     Si Vérificateur( $G, k, D$ ) = Vrai alors
5:       Retourner  $D$ 
6:     Fin Si
7:   Fin Pour
8: Fin Pour
9: Retourner  $\emptyset$ 

```

L'algorithme génère tous les sous-ensembles de sommets de taille au plus k , et il y en a 2^k . Pour chaque sous-ensemble, l'algorithme appelle Vérificateur dont la complexité est $O(|V| + |E|)$. La complexité de l'algorithme est donc $O(2^k \cdot (|V| + |E|))$, ce qui est exponentiel en la taille du graphe.

Question 5. Le vérificateur prend en entrée l'instance (ici un graphe G et un entier k) ainsi qu'un certificat. Le certificat est un sous-ensemble de sommets, il s'agit d'une solution potentielle du problème EDNA. Le vérificateur renvoie vrai si et seulement si le certificat est bien une solution de EDNA. Sa complexité est polynomiale en la taille du graphe.

L'algorithme de la question 4 ne prend en entrée qu'une instance de EDNA, c'est-à-dire un graphe G et un entier k . Il renvoie un ensemble dominant non-ambigu de taille au plus k si G en possède un, et l'ensemble vide sinon. Pour cela, il énumère toutes les solutions possibles du problème EDNA, c'est-à-dire tous les sous-ensembles de sommets de G de taille au plus k (ceux de taille $> k$ ne nous intéressent pas), et appelle le vérificateur pour déterminer s'il s'agit bien d'un ensemble non-ambigu ou non. Sa complexité est exponentielle en la taille de l'instance. Comme l'algorithme génère toutes les solutions possibles, on est certains que si G admet un ensemble dominant non-ambigu de taille au plus k , alors l'algorithme le trouvera.

Question 6. Soit $G = (V, E)$ un graphe, et k un entier. Ainsi, $\langle G, k \rangle$ est une instance de EDNA. On construit à partir de $\langle G, k \rangle$ une formule ϕ_G instance de SAT. On montre ensuite que $\langle G, k \rangle$ possède un ensemble-dominant non-ambigu de taille au plus k si et seulement si ϕ_G est satisfaisable. Notons dans ce qui suit $n = |V|$ et $m = |E|$.

Pour chaque sommet u de G , on crée k variables $x_{u,1}, x_{u,2}, \dots, x_{u,k}$. La formule possède ainsi $n \times k$ variables. On définit maintenant les clauses de ϕ_G . Tout d'abord, il faut exprimer que l'on souhaite un sous-ensemble de sommets de taille au plus k . Soit $X = \{l_1, l_2, \dots, l_{|X|}\}$ un sous-ensemble de variables de ϕ_G . On définit la sous-formule $Au_plus(X)$ qui garantit qu'au plus une variable de l'ensemble X sera vraie :

$$Au_plus(X) := \bigwedge_{1 \leq i < j \leq |X|} (\neg l_i \vee \neg l_j)$$

On utilise cette formule pour s'assurer que pour tout $1 \leq i \leq k$, au plus un sommet est le i -ème sommet de l'ensemble. On obtient la formule ψ_1 suivante :

$$\psi_1 := \bigwedge_{1 \leq i \leq k} Au_plus(\{x_{u,i} \mid u \in V\})$$

On définit maintenant une sous-formule ψ_2 afin de s'assurer qu'un sommet u ne peut pas être à la fois le i -ème sommet et le j -ème sommet (avec $i \neq j$) de D :

$$\psi_2 := \bigwedge_{u \in V} Au_plus(\{x_{u,i} \mid 1 \leq i \leq k\})$$

Il nous reste maintenant à vérifier que l'ensemble construit est bien un ensemble dominant non-ambigu, c'est à dire que chaque sommet de G est dominé exactement une fois. Pour cela, on définit la sous-formule $Au_moins(X)$ qui permet de s'assurer qu'au moins une variable de X sera vraie :

$$Au_moins(X) := \bigvee_{1 \leq i \leq |X|} l_i$$

Nous pouvons maintenant définir la formule ψ_3 qui garantit que chaque sommet est dominé exactement une fois par l'ensemble :

$$\psi_3 := \bigwedge_{u \in V} Au_moins(\{x_{v,i} \mid v \in N[u], 1 \leq i \leq k\}) \wedge Au_plus(\{x_{v,i} \mid v \in N[u], 1 \leq i \leq k\})$$

La formule finale ϕ_G est donc $\phi_G := \psi_1 \wedge \psi_2 \wedge \psi_3$. Notons qu'elle est en forme normale conjonctive.

Question 7. On montre que la réduction est bien valide. Supposons d'abord que $\langle G, k \rangle$ est une instance positive, et soit D un ensemble dominant non-ambigu de G . On numérote arbitrairement les sommets de D : $v_1, v_2, \dots, v_{|D|}$, avec $|D| \leq k$. On construit alors une assignation des variables de ϕ_G : on met la variable $x_{u,i}$ à **Vrai** si et seulement si u est le i -ème sommet de D , i.e. $u = v_i$. Ainsi, exactement $|D| \leq k$ variables sont vraies, ce qui satisfait ψ_1 . Comme tous les sommets de D sont deux à deux différents, ψ_2 est satisfaite. Enfin, ψ_3 est satisfaite car D est bien un ensemble dominant non-ambigu, i.e. chaque sommet de G est dominé exactement une fois par D .

Supposons maintenant que ϕ_G est satisfaisable et considérons une assignation des variables de ϕ_G qui satisfait la formule. La formule ψ_1 nous assure que au plus k variables sont vraies puisque que pour tout $1 \leq i \leq k$, au plus une variable $x_{u,i}$ est vraie. La formule ψ_2 garantit que l'ensemble des variables qui sont vraies correspondent à des sommets de G différents. Ainsi, soit $D := \{u \mid \exists i \in [1, k] \text{ tel que } x_{u,i} \text{ est vraie}\}$. Ainsi, $|D| \leq k$ et ψ_3 assure que D est bien un ensemble dominant non-ambigu.

Question 8. Rappelons que la formule ϕ_G contient $n \times k$ variables et donc que le nombre de littéraux de ϕ_G correspond à $2 \times n \times k$. Regardons maintenant la complexité de cette réduction. La formule $Au_plus(X)$ teste toutes les paires de X , soit $\frac{|X| \times (|X|-1)}{2} = O(|X|^2)$. Ainsi, la formule ψ_1 prend un temps $O(k \times n^2)$. La formule ψ_2 prend un temps $O(n \times k^2)$.

Enfin, l'analyse de la complexité de la formule ψ_3 est un peu plus complexe. Rappelons que $|N[u]| = |Adj[u]| + 1 = \deg(u) + 1$, où $\deg(u)$ est le degré de u dans G . Pour chaque sommet, ψ_3 "appelle" Au_moins sur l'ensemble $X_u = \{x_{v,i} \mid v \in N[u], 1 \leq i \leq k\}$ de taille $k \times (\deg(u) + 1)$. Comme pour tout graphe G , la somme des degrés vaut deux fois le nombre d'arêtes, la formule Au_plus coûte $O(k \times (n + m))$.

Elle "appelle" ensuite pour chaque sommet u la formule Au_plus sur l'ensemble X_u de taille $k \times (\deg(u) + 1)$. Comme $\deg(u) \leq n$, $|X_u| = O(k \times n)$. La formule ψ_3 coûte donc en appelant Au_plus pour chaque sommet $n \times O((k \times n)^2) = n \times O(k^2 \times n^2)$, ce qui donne $O(k^2 n + n^3)$ au total. Ainsi, la formule ψ_3 coûte $O(k \times (n + m) + O(k^2 n + n^3))$.

Ainsi, ϕ_G est construite en temps $O(k \times n^2) + O(n \times k^2) + O(k \times (n + m)) + O(k^2 n + n^3) = O(n^3)$ car $k = O(n)$. La complexité de la réduction est donc polynomiale en la taille du graphe.

Question 9. Dans les questions 6, 7 et 8, nous avons démontré que EDNA se réduit polynomialement au problème SAT. Comme SAT est dans NP, nous en déduisons que EDNA appartient à NP.